Querying Metrics from Prometheus

🕐 2 minute read   ✔ page test

This task shows you how to query for Istio Metrics using Prometheus. As part of this task, you will use the web-based interface for querying metric values.

The Bookinfo sample application is used as the example application throughout this task.

Before you begin

- Install Istio in your cluster.
- Install the Prometheus Addon.
- Deploy the Bookinfo application.

Querying Istio metrics

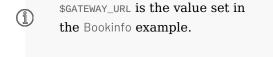1. Verify that the prometheus service is running in your cluster.

   In Kubernetes environments, execute the following command:

   ```
   $ kubectl -n istio-system get svc prometheus
   NAME         TYPE         CLUSTER-IP       EXTER
   NAL-IP   PORT(S)   AGE
   prometheus   ClusterIP    10.109.160.254   <none
   >        9090/TCP   4m
   ```

2. Send traffic to the mesh.

   For the Bookinfo sample, visit `http://$GATEWAY_URL/productpage` in your web browser or issue the following command:

   ```
   $ curl "http://$GATEWAY_URL/productpage"
   ```

   > ⓘ $GATEWAY_URL is the value set in the Bookinfo example.

3. Open the Prometheus UI.

   In Kubernetes environments, execute the following command:

   ```
   $ istioctl dashboard prometheus
   ```

   Click **Graph** to the right of Prometheus in the header.

4. Execute a Prometheus query.

   In the "Expression" input box at the top of the web page, enter the text:

   ```
   istio_requests_total
   ```
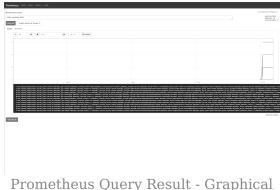
   Then, click the **Execute** button.

The results will be similar to:

Prometheus Query Result

You can also see the query results

graphically by selecting the Graph tab underneath the **Execute** button.

Prometheus Query Result - Graphical

Other queries to try:

- Total count of all requests to the productpage service:

```
istio_requests_total{destination_service="produ
ctpage.default.svc.cluster.local"}
```

- Total count of all requests to `v3` of the reviews service:

```
istio_requests_total{destination_service="revie
ws.default.svc.cluster.local", destination_vers
ion="v3"}
```

  This query returns the current total count of all requests to the v3 of the reviews service.

- Rate of requests over the past 5 minutes to all instances of the productpage service:

```
rate(istio_requests_total{destination_service=~
"productpage.*", response_code="200"}[5m])
```

About the Prometheus

addon

The Prometheus addon is a Prometheus server that comes preconfigured to scrape Istio endpoints to collect metrics. It provides a mechanism for persistent storage and querying of Istio metrics.

For more on querying Prometheus, please read their querying docs.

Cleanup

- Remove any istioctl processes that may still be running using control-C or:

  ```
  $ killall istioctl
  ```

- If you are not planning to explore any

follow-on tasks, refer to the Bookinfo cleanup instructions to shutdown the application.